CLAIMS

What is claimed is:

1. A method of generating a document, the method comprising:

establishing an architecture for a set of rules to be used in documents that consist of a plurality of components; and

creating a dynamic document structure that can resolve to one or more instances of a document and that is configured to include one or more rules based on the architecture for a set of rules.

2. A method as claimed in claim 1, wherein establishing an architecture for a set of rules includes creating a schema having a conditions element.

3. A method as claimed in claim 1, wherein establishing an architecture for a set of rules includes creating a schema having a choose element.

4. A method as claimed in claim 1, wherein establishing an architecture for a set of rules rules includes creating a schema having an iterators element.

5. A method as claimed in claim 1, wherein establishing an architecture for a set of rules includes creating a schema having a functions element.

6. A method as claimed in claim 1, wherein establishing an architecture for a set of rules includes creating a schema having a conditions element, a choose element, an iterators element, and a functions element.

7. A method as claimed in claim 1, wherein establishing an architecture for a set of rules includes creating a schema having an external interface element that is configured to be resolved into a value.

8. A method as claimed in claim 7, wherein the value is chosen from a group that includes a set, an XML DOM node, and an XML DOM node list.

9. A method as claimed in claim 7, wherein the external data interface element is configured to have an entity reference attribute.

10. A method as claimed in claim 7, wherein the external data interface element is configured to have a return type attribute.

11. A method as claimed in claim 1, wherein establishing an architecture for a set of rules includes creating a schema having an internal interface element that is configured specify the usage of data resolved by an external interface element.

12. A method as claimed in claim 1, further comprising creating a static document structure that can be resolved into one or more instances of a document that includes at least some content that is determined before and some content that is unchanged during and after a resolution process.

13. A method as claimed in claim 1, further comprising providing a data set configured to be processable by one or more rules built on the architecture for a set of rules.

14. A method of generating a document, the method comprising:

establishing an architecture for a set of rules by creating a schema having a conditions element, a choose element, an iterators element, a functions element, and an external interface element that is configured to be resolved into a value; and

creating a dynamic document structure that can resolve to one or more instances of a document using the set of rules.

15. A method of generating a document, the method comprising:

establishing an architecture for a set of rules including a conditions element, a choose element, an iterators element, and a functions element;

creating a dynamic document structure that can resolve to one or more instances of a document using the set of rules; and creating a static document structure that can be resolved into one or more instances of a document that includes at least some content that is determined before and some content that is unchanged during and after a resolution process.

16. A method of assembling a document from a group of components, the method comprising:

creating a transaction data set;

retrieving one or more cross-referenced document components from a data base based on the transaction data set, the one or more document components configured to include one or more rules;

processing the one or more cross-referenced document components in a processor to generate a tree having a root node;

processing the tree beginning at the root node; and

when a rule is encountered, evaluating the rule and replacing it with a value;

17. A method as claimed in claim 16, further comprising establishing an architecture for a set of rules.

18. A method as claimed in claim 17, wherein establishing an architecture for a set of rules includes creating a schema having a conditions element.

19. A method as claimed in claim 17, wherein establishing an architecture for a set of rules includes creating a schema having a choose element.

20. A method as claimed in claim 17, wherein establishing an architecture for a set of rules includes creating a schema having an iterators element.

21. A method as claimed in claim 17, wherein establishing an architecture for a set of rules includes creating a schema having a functions element.

22. A method as claimed in claim 17, wherein establishing an architecture for a set of rules includes creating a schema having a conditions element, a choose element, an iterators element, and a functions element.

23. A method as claimed in claim 17, wherein establishing an architecture for a set of rules includes creating a schema having an external interface element that is configured to be resolved into a value.

24. A method as claimed in claim 23, wherein the value is chosen from a group that includes a set, an XML DOM node, and an XML DOM node list.

25. A method as claimed in claim 23, wherein the external data interface element is configured to have an entity reference attribute.

26. A method as claimed in claim 23, wherein the external data interface element is configured to have a return type attribute.

27. A method as claimed in claim 17, wherein establishing an architecture for a set of rules includes creating a schema having an internal interface element that is configured specify the usage of data resolved by an external interface element.

28. A method of assembling a data structure from a group of components, the method comprising:

creating a transaction data set;

retrieving one or more cross-referenced data structure components from a database based on the transaction data set, the one or more data structure components configured to include one or more rules;

processing the one or more cross-referenced data structure components in a processor to generate a tree having a root node;

processing the tree beginning at the root node; and

when a rule is encountered, evaluating the rule and replacing it with a value.

29. A method as claimed in claim 28, further comprising establishing an architecture for a set of rules.

30. A method as claimed in claim 28, further comprising establishing a list of data structures and performing each of the steps in claim 28 for each of the data structures.